

Приложение 1  
к комплекту оценочных материалов по  
программе 09.02.07 Информационные системы и  
программирование

## ТЕСТОВЫЕ ЗАДАНИЯ

### **ПК-1.5 Осуществлять рефакторинг и оптимизацию программного кода**

#### **Задание 1 (Основы алгоритмизации и программирования)**

*Прочтите текст, выберите правильный ответ.*

**Какой этап жизненного цикла программного обеспечения посвящен непосредственной разработке и написанию программного кода?**

1. Документирование
2. Кодирование
3. Анализ и проектирование
4. Тестирование

Ответ: \_\_\_\_\_

*Ключ ответа: 2*

#### **Задание 2 (Поддержка и тестирование программных модулей)**

*Прочтите текст, выберите правильный ответ.*

**Какой инструмент используется для статического анализа кода и выявления потенциальных проблем в проекте JavaScript?**

1. Express.js
2. ESLint
3. Webpack
4. Prettier

Ответ: \_\_\_\_\_

*Ключ ответа: 2*

#### **Задание 3 (Численные методы)**

*Прочтите текст, выберите правильный ответ.*

**Какой метод обычно используется для решения нелинейных уравнений?**

1. Метод Гаусса
2. Метод Ньютона (касательных)
3. Метод Якоби
4. Метод прямоугольников

Ответ: \_\_\_\_\_

*Ключ ответа: 2*

#### **Задание 4 (Поддержка и тестирование программных модулей)**

*Прочтите текст, выберите правильный ответ.*

**Какой тип тестирования проверяет приложение на устойчивость к нагрузкам и способности выдерживать высокие объемы обращений пользователей?**

1. Регрессионное тестирование
2. Нагрузочное тестирование
3. Интеграционное тестирование
4. А/В-тестирование

Ответ: \_\_\_\_\_

*Ключ ответа: 2*

### **Задание 5 (Основы алгоритмизации и программирования)**

*Прочтите текст, выберите правильный ответ.*

**Какой оператор в Python используется для приведения строки к целому числу?**

1. int()
2. float()
3. str()
4. bool()

Ответ: \_\_\_\_\_

*Ключ ответа: 1*

### **Задание 6 (Поддержка и тестирование программных модулей)**

*Прочтите текст, выберите правильные ответы.*

**Какие утверждения о юнит-тестировании являются верными?**

1. Юнит-тесты проверяют небольшие фрагменты кода отдельно от остальной системы.
2. Юнит-тесты должны охватывать полный цикл взаимодействия пользователя с интерфейсом.
3. Каждый юнит-тест проверяет отдельный аспект функциональности.
4. Юнит-тесты требуют наличия реальных внешних зависимостей, таких как база данных или API.
5. Юнит-тесты помогают выявить баги на ранних стадиях разработки.
6. Юнит-тесты используются только для обнаружения ошибок в готовой программе.

Ответ: \_\_\_\_\_

*Ключ ответа: 135*

### **Задание 7 (Основы алгоритмизации и программирования, Производственная практика (преддипломная))**

*Прочтайте текст, заполните пропуски.*

1. Одним из ключевых мотивов для рефакторинга является устранение \_\_\_\_\_ — повторения одинаковых или схожих фрагментов кода в разных местах программы.
2. Для выявления «узких мест» производительности перед оптимизацией следует использовать \_\_\_\_\_, который покажет, какие участки кода тратят больше всего времени.
3. Снижение \_\_\_\_\_ функции (количество ветвлений) делает код проще для понимания и тестирования.
4. Оптимизация часто начинается с устранения \_\_\_\_\_ — вычислений, которые выполняются многократно без необходимости или дают уже известный результат.
5. Важная цель рефакторинга — повышение \_\_\_\_\_, чтобы другие разработчики могли легко разбираться в коде и поддерживать его.

Ответ: \_\_\_\_\_

*Ключ ответа: 1.дублирование; 2.профилировщик; 3.сложность; 4.избыточность; 5.читаемость.*

### **Задание 8 (Поддержка и тестирование программных модулей)**

*Прочтите текст, выберите правильные ответы.*

**Какие из перечисленных методов используются для создания тестовых данных?**

1. Ручной ввод данных для каждого теста
2. Использование фабрик объектов (Object Mother pattern)
3. Генерация случайных тестовых данных
4. Использование фикстур (fixtures)

5. Копирование данных из продакшн-окружения
6. Использование конструкторов с параметрами по умолчанию

Ответ: \_\_\_\_\_

Ключ ответа: 2346

### **Задание 9 (Основы алгоритмизации и программирования, ГИА)**

*Прочтите текст, выберите правильные ответы.*

**Какие техники помогают создать надежные и устойчивые тестовые наборы данных для тестирования программного обеспечения?**

1. Генерация тестовых данных вручную для каждого сценария
2. Генерация тестовых данных с помощью специализированных генераторов
3. Использование паттерна Object Mother для создания фабричных классов

данных

4. Применение фикстур (fixtures) для подготовки стандартного набора данных
5. Копирование данных прямо из рабочей среды
6. Автоматическая генерация тестов с использованием случайно созданных данных

Ответ: \_\_\_\_\_

Ключ ответа: 135

### **Задание 10 (Поддержка и тестирование программных модулей,**

**Производственная практика (преддипломная)**

*Прочтите текст и запишите ответ.*

**Сколько возможных комбинаций настроек может быть создано, если пользователю предлагается настроить три независимых переключателя, каждый из которых может находиться в двух положениях (включён или выключен)?**

Ответ: \_\_\_\_\_

Ключ ответа:

Эталонный ответ 1. 8

Эталонный ответ 2. 8 комбинаций

### **Задание 11 (Основы алгоритмизации и программирования, Учебная практика**

**ПМ.01 Разработка модулей программного обеспечения для компьютерных систем)**

*Прочтайте текст и установите последовательность.*

**Расположите этапы методики Test Driven Development (TDD) в правильной последовательности.**

1. Написание минимального количества кода для прохождения теста
2. Рефакторинг кода
3. Написание нового failing-теста
4. Повторение цикла для новой функциональности
5. Запуск тестов и проверка, что новый тест не проходит

Ответ: \_\_\_\_\_

Ключ ответа: 35124

### **Задание 12 (Основы алгоритмизации и программирования)**

*Прочтите текст, выберите правильный ответ.*

**Какой принцип ООП предполагает, что класс может наследовать свойства и методы другого класса?**

1. Инкапсуляция
2. Полиморфизм
3. Наследование
4. Абстракция

Ответ: \_\_\_\_\_

Ключ ответа: 3

### **Задание 13 (Поддержка и тестирование программных модулей)**

*Прочтите текст, установите соответствие.*

<b>Задачи</b>		<b>Инструменты</b>	
A	Выявить участки кода, потребляющие наибольшее время выполнения	1.	Профилизатор (например, Visual Studio Profiler, Valgrind).
B	Устранить повторяющиеся фрагменты кода в разных частях программы	2.	Статический анализатор кода (например, SonarQube, Pylint).
B	Уменьшить количество условных операторов и вложенных циклов в функции	3.	Вынесение повторяющегося кода в отдельную функцию или метод.
		4.	Система контроля версий (Git).
		5.	Рефакторинг с упрощением логики (замена сложных условий на более понятные конструкции, использование полиморфизма и т. п.).

*Запишите выбранные цифры под соответствующими буквами.*

A	Б	В

*Ключ ответа: A1 Б3 В5*

### **Задание 14 (Основы алгоритмизации и программирования, Производственная практика ПМ.01 Разработка модулей программного обеспечения для компьютерных систем)**

*Прочтите текст, выберите правильные ответы.*

**Какие из перечисленных утверждений о рефакторинге являются верными?**

1. Рефакторинг не должен изменять внешнее поведение программы
2. Рефакторинг проводят, чтобы ускорить выполнение программы
3. Рефакторинг улучшает читаемость и поддерживаемость кода
4. Рефакторинг обычно выполняется без наличия тестов
5. Выделение метода — это пример приема рефакторинга
6. Рефакторинг проводят только перед релизом продукта

Ответ:

*Ключ ответа: 135*

### **Задание 15 (Численные методы)**

*Прочтайте текст, выберите правильный ответ.*

**Какой метод используется для численного интегрирования функции?**

1. Метод Ньютона
2. Метод Гаусса
3. Метод бисекции
4. Метод Симпсона

Ответ:

*Ключ ответа: 4*

### **Задание 16 (Поддержка и тестирование программных модулей, Производственная практика (преддипломная))**

*Прочтите текст, установите последовательность.*

**Расположите этапы процесса рефакторинга и оптимизации кода в правильной последовательности — от начальной диагностики до завершения работ.**

1. Внесение изменений в код: упрощение алгоритмов, устранение дублирования, оптимизация структур данных.

2. Фиксация исходного состояния: замер времени выполнения и потребления памяти до оптимизации.

3. Проверка работоспособности после изменений — запуск юнит-тестов и интеграционных тестов.

4. Анализ кода с помощью профилировщика для выявления «узких мест» (горячих точек).

5. Повторный замер производительности для оценки эффекта от внесённых изменений.

Ответ: \_\_\_\_\_  
Ключ ответа: 24135

### Задание 17 (Основы алгоритмизации и программирования, ГИА)

Прочтите текст и запишите ответ.

Как называется принцип, при котором класс должен быть открыт для расширения, но закрыт для модификации? (Ответ введите на русском или английском языке)

Ответ: \_\_\_\_\_  
Ключ ответа: Принцип открытости/закрытости (Open/Closed Principle)

### Задание 18 (Поддержка и тестирование программных модулей, Учебная практика ПМ.01 Разработка модулей программного обеспечения для компьютерных систем)

Прочтите текст, установите соответствие.

Установите соответствие между метрикой тестирования и её описанием.

Метрика тестирования		Описание метрики	
A	Покрытие кода (Code coverage)	1.	Количество линейно независимых путей в программе
Б	Цикломатическая сложность (Cyclomatic complexity)	2.	Время выполнения программы
В	Технический долг (Technical debt)	3.	Стоимость последующего переписывания кода
		4.	Процент кода, который выполняется при прохождении тестов
		5.	Количество строк кода

Запишите выбранные цифры под соответствующими буквами.

A	Б	В

Ключ ответа: А4 Б1 В3

### Задание 19 (Основы алгоритмизации и программирования)

Прочтите текст и установите последовательность.

Установите правильную последовательность этапов выполнения программы, написанной на Java.

1. Компиляция в байт-код
2. Написание исходного кода
3. Интерпретация или JIT-компиляция JVM
4. Выполнение программы
5. Сборка проекта (build)

Ответ: \_\_\_\_\_

Ключ ответа: 25134

### Задание 20 (Поддержка и тестирование программных модулей, Производственная практика ПМ.01 Разработка модулей программного обеспечения для компьютерных систем)

*Прочтите текст, выберите правильные ответы.*

**Какие из перечисленных практик способствуют повышению качества кода?**

1. Регулярный рефакторинг
2. Коллективное владение кодом (Collective Code Ownership)
3. Написание комментариев для каждого метода
4. Проведение код-ревью
5. Использование длинных и описательных имен переменных
6. Сокрытие всех полей класса через модификатор `private`

Ответ:

*Ключ ответа: 1245*

---